

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «САМАРСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИМЕНИ АКАДЕМИКА С.П. КОРОЛЕВА  
(САМАРСКИЙ УНИВЕРСИТЕТ)»

Институт Информатики и кибернетики   
Кафедра Программных систем

**ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**  
  
к лабораторной работе №4 по дисциплине «Прикладные информационные системы»

Обучающийся группы 6132-020402D \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Дьяконов А.В.

Обучающийся группы 6132-020402D \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Лазарев М.Ю.

Руководитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Лобанков А.А.

Самара 2024

СОДЕРЖАНИЕ

[1 Постановка задачи 3](#_Toc182534592)

[2 Результаты работы 4](#_Toc182534593)

[2.1 Результаты задания 1 4](#_Toc182534594)

[2.2 Результаты задания 2 5](#_Toc182534595)

[2.3 Результаты задания 3 7](#_Toc182534596)

[2.4 Результаты задания 4 8](#_Toc182534597)

[2.5 Результаты задания 5 8](#_Toc182534598)

[ПРИЛОЖЕНИЕ А Листинг программы 11](#_Toc182534599)

1. Постановка задачи

В процессе выполнения заданий ознакомиться с основами Angular.

1. Результаты работы
   1. Результаты задания 1

Создайте Angular проект.

1. В терминале выполняем команду npm install -g @angular/cli.
2. Открываем терминал PowerShell в нашей созданной папке будущего проекта и вводим ng new digital-department-application.
3. Открываем в VS Code папку проекта digital-department-application и вводим в терминале VS Code npm run start.
4. Получаем результат (Рис. 1) и открываем в браузере http://localhost:4200 (Рис. 2).

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

Рисунок 1 – Результат работы терминала

Изображение выглядит как текст, снимок экрана, программное обеспечение, мультимедиа

Автоматически созданное описание

Рисунок 2 – Приложение Angular в браузере

* 1. Результаты задания 2

Добавьте 3 компонента - страницы:

- О себе

- Приветствие

- Группы

На странице «О себе» добавьте информацию о себе слева на странице (используйте для этого различные html теги: абзац, заголовки, нумерованные и маркированные списки), справа добавьте фотографию, чтобы это выглядело как две карточки на странице. Внизу страницы добавьте кнопку «Перейти к моей группе», по клику на которой будет осуществлен переход на страницу «Группы». Номер вашей группы будет передан в качестве параметра маршрутизации. Стилизуйте страницу. На странице «Группы» должна выводиться информация о номере группы, если номер передан в качестве параметра маршрутизации, в противном случае должна выводиться надпись: «Группа не найдена».

1. Создаем компоненты «О себе», «Приветствие» и «Группы», прописываем ng generate component about, ng generate component greet и ng generate component group (соответственно названиям) в терминале VS Code для создания шаблонов. Получаем независимые компоненты (параметр standalone: true) и в основной ts-файл каждого компонента из @angular/router импортируем RouterModule для дальнейшей организации роутинга.
2. Стилизуем страницу «О себе» (Приложение А). Редактируем компонентные css и html файлы. В TS файле в экспортированном классе AboutComponent добавляем параметр router и прописываем метод goToGroup, который отправляет на страницу «Группы» при помощи router.navigate по номеру нашей группы 6132. В браузере получаем итоговый вариант перейдя по http://localhost:4200/about (Рис.3)(Рис.4) и по клику «Перейти к моей группе» (Рис.5).

Изображение выглядит как текст, Человеческое лицо, человек, снимок экрана

Автоматически созданное описание

Рисунок 3 – Страница «О себе»

Изображение выглядит как текст, Человеческое лицо, человек, снимок экрана

Автоматически созданное описание

Рисунок 4 – Страница «О себе»

Изображение выглядит как текст, снимок экрана, программное обеспечение

Автоматически созданное описание

Рисунок 5 – Результат перехода по клику «Перейти к моей группе»

1. Реализуем страницу «Группы» (Приложение А). Редактируем html-файл, который выводит на странице «Группа не найдена» при переходе без параметра маршрутизации. В ts-файле прописываем параметры для класса GroupComponent, который в свою очередь реализует OnInit из @angular/core. В качестве route передаётся ActivatedRoute, а в параметрах класса добавляем номер группы groupNumber, Передачу номера группы по параметру маршрутизации, реализуем с помощью ngOnInit(). В итоге получаем результат с передачей параметра маршрутизации (Рис.5) и переходу по прямой ссылке http://localhost:4200/group (Рис.6) в браузере.

Изображение выглядит как текст, снимок экрана, Шрифт, программное обеспечение

Автоматически созданное описание

Рисунок 6 – Страница «Группы»

* 1. Результаты задания 3

Создайте навигационную панель для перемещения по страницам в виде отдельного компонента. Добавьте эту панель на все страницы.

1. Создаём компонент для нашей панели навигации, вводим ng generate component navigation в терминале VS Code для создания шаблонов. Редактируем в папке app/navigation ts-файл, импортируем RouterModule для будущей реализации роутинга. Редактируем css-файл для стилизации нашей app-navigation панели и прописываем в html-файле ссылки routerLink на все наши страницы, а так же определяем, что это навигационная панель navbar (Приложение А).
2. В файле app.component.html для отображения панели на всех страницах прописываем заголовок нашего приложения, навигационную панель app-navigation и router-outlet. В app.component.ts импортируем RouterModule, RouterOutlet и компонент навигационной панели NavigationComponent, задаем заголовок AngularLab (Приложение А).
3. Видим результат в браузере (Рис.7).

Изображение выглядит как текст, снимок экрана, Шрифт, Цвет электрик

Автоматически созданное описание

Рисунок 7 – Навигационная панель

* 1. Результаты задания 4

Настройте роутинг для ваших страниц. Если идет попытка перехода на несуществующий url, то пользователя должно перекинуть на страницу «Приветствие».

1. В app.routes.ts импортируем все наши компоненты и в классе routes указываем параметры path, соответствующие нашим компонентам. Реализуем дополнительно переход на страницу группы с наличием параметра маршрутизации :groupNumber и переход redirectTo на страницу «Приветствие» при указании несуществующего URL «\*\*» (Приложение А).
   1. Результаты задания 5

На странице «Приветствие» реализуйте запрос имени пользователя и отображение приветствия пользователю по нажатию на кнопку. Продемонстрируйте работу жизненного цикла компонента. Имплементируйте все методы жизненного цикла.

1. Стилизуем страницу «Приветствие» (Приложение А). Прописываем в css-файл пропишем стиль нашей кнопки и поля ввода. В html-файле пропишем вид нашей страницы с указанием нашего greeting-container, ввод с помощью ngModel нашего имени пользователя userName, реализацию клика по кнопке «Say Hello» функции showGreeting() и вывода приветствия с помощью ngif приветствия greeting.
2. Реализацию всего функционала и жизненного цикла прописываем в ts-файле, где мы импортируем из @angular/core функции жизненного цикла с соответствующими вызовами. В экспортируемом классе GreetComponent прописываем сохранение имени пользователя userName и текст приветствия greeting в качестве типа string. Наш класс GreetComponent имплементирует и вызывает при инициализации компонента ngOnInit(), при изменении входных свойств ngOnChanges(), при каждой проверке изменений ngDoCheck(), после инициализации контента ngAfterContentInit(), после каждой проверки контента ngAfterContentChecked(), после инициализации представления ngAfterViewInit(), после каждой проверки представления ngAfterViewChecked(), при уничтожении компонента ngOnDestroy(), а также содержит метод для отображения приветствия showGreeting(). В браузере получаем итоговые результаты - страницу приветствия(Рис.8)(Рис.9) и работу жизненного цикла компонента в консоли(Рис.10)(Рис.11).

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

Рисунок 8 – Страница «Приветствие»

Изображение выглядит как текст, снимок экрана, программное обеспечение, Значок на компьютере

Автоматически созданное описание

Рисунок 9 – Ввод имени

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Рисунок 10 – Жизненный цикл компонента

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

Рисунок 11 – Закрытие компонента

ПРИЛОЖЕНИЕ А  
Листинг программы

#app/about/about.component.html

<div class="about-container">

  <!-- Первая карточка с текстом -->

  <div class="about-card">

    <h1>О себе - Лазарев Михаил</h1>

    <p>Меня зовут Михаил. Я учусь в магистратуре ФИИТ СГАУ на кафедре Программных систем.</p>

    <h2>Интересы:</h2>

<ul>

<li>Тестирование</li>

<li>Геймдев</li>

<li>Искусственный интеллект</li>

</ul>

<h2>Опыт:</h2>

<ol>

<li>Стажировка QA Gamedev</li>

<li>Курс Big Data</li>

<li>Опенсорс</li>

</ol>

</div>

<div class="about-photo">

    <img src="assets/person1-photo.jpg" alt="Фотография" />

  </div>

  <div class="about-card">

    <h1>О себе - Дьяконов Александр</h1>

    <p>Меня зовут Александр. Я учусь в магистратуре ФИИТ СГАУ на кафедре Программных систем.</p>

<h2>Интересы:</h2>

    <ul>

      <li>Бизнес-аналитика</li>

      <li>Программная инженерия</li>

      <li>IT-Технологии</li>

    </ul>

    <h2>Опыт:</h2>

    <ol>

      <li>Буткэмп</li>

      <li>Обучение бизнес анализу</li>

      <li>Работа бизнес-аналитиком</li>

    </ol>

  </div>

  <div class="about-photo">

    <img src="assets/person2-photo.jpg" alt="Фотография" />

  </div>

</div>

<button (click)="goToGroup()">Перейти к моей группе</button>

#app/about/about.component.css

.about-container {

  display: grid;

  grid-template-columns: repeat(2, 1fr);

  gap: 20px;

  padding: 20px;

}

.about-card {

  background-color: #539ff123;

  padding: 20px;

  border-radius: 8px;

}

.about-photo {

  display: flex;

  justify-content: center;

  align-items: center;

  background-color: #539ff123;

  padding: 20px;

  border-radius: 8px;

}

.about-photo img {

  width: 300px;

  height: 300px;

  object-fit: cover;

  border-radius: 8px;

}

button {

  margin-top: 20px;

  padding: 10px 15px;

  background-color: #539ff123;

  color: rgb(2, 2, 2);

  border: none;

  border-radius: 5px;

  cursor: pointer;

}

button:hover {

  background-color: #0056b3;

}

#app/about/about.component.ts

import { Component } from '@angular/core';

import { Router, RouterModule } from '@angular/router';

@Component({

  selector: 'app-about',

  templateUrl: './about.component.html',

  standalone: true,

  imports: [RouterModule],

  styleUrls: ['./about.component.css']

})

export class AboutComponent {

  constructor(

    private router: Router

              ) {}

  goToGroup() {

    const groupNumber = '6132';

    this.router.navigate(['/group', groupNumber]);

  }

}

#app/group/group.component.html

<div class="group-info">

<h1>Информация о группе</h1>

<p \*ngIf="groupNumber; else noGroup">Номер группы: {{ groupNumber }}</p>

<ng-template #noGroup>

<p>Группа не найдена</p>

</ng-template>

</div>

#app/group/group.component.ts

import { Component, OnInit } from '@angular/core';

import { ActivatedRoute, RouterModule } from '@angular/router';

import { CommonModule } from '@angular/common';

@Component({

selector: 'app-groups',

templateUrl: './group.component.html',

imports: [CommonModule, RouterModule],

standalone: true,

styleUrls: ['./group.component.css']

})

export class GroupComponent implements OnInit {

groupNumber: string | null = null;

constructor(private route: ActivatedRoute) {}

ngOnInit(): void {

this.route.paramMap.subscribe(params => {

this.groupNumber = params.get('groupNumber');

});

}

}

# app/navigation/navigation.component.ts

import { Component } from '@angular/core';

import { RouterOutlet, RouterModule } from '@angular/router';

import { NavigationComponent } from './navigation/navigation.component';

@Component({

  selector: 'app-root',

  templateUrl: './app.component.html',

  styleUrl: './app.component.css',

  standalone: true,

  imports: [RouterOutlet, NavigationComponent, RouterModule]

})

export class AppComponent {

  title = 'AngularLab';

}

# app/navigation/navigation.component.css

.navbar {

    display: flex;

    gap: 20px;

    background-color: #007bff;

    padding: 10px;

  }

  .navbar a {

    color: white;

    text-decoration: none;

    font-weight: bold;

  }

  .navbar a.active {

    text-decoration: underline;

  }

# app/navigation/navigation.component.html

<nav class="navbar">

    <a routerLink="/about">О себе</a>

    <a routerLink="/greet">Приветствие</a>

    <a routerLink="/group">Группы</a>

  </nav>

#app/app.component.html

<h1>{{title}}</h1>

<app-navigation></app-navigation>

<router-outlet></router-outlet>

#app/app.component.ts

import { Component } from '@angular/core';

import { RouterOutlet, RouterModule } from '@angular/router';

import { NavigationComponent } from './navigation/navigation.component';

@Component({

  selector: 'app-root',

  templateUrl: './app.component.html',

  styleUrl: './app.component.css',

  standalone: true,

  imports: [RouterOutlet, NavigationComponent, RouterModule]

})

export class AppComponent {

  title = 'AngularLab';

}

#app/app.routes.ts

import { Routes } from '@angular/router';

import {GreetComponent} from "./greet/greet.component";

import {AboutComponent} from "./about/about.component";

import {GroupComponent} from "./group/group.component";

export const routes: Routes = [

    { path: "greet", component: GreetComponent},

    { path: "about", component: AboutComponent},

    { path: "group", component: GroupComponent},

    { path: "group/:groupNumber", component: GroupComponent},

    { path: "\*\*", redirectTo: "/greet", pathMatch:"full"}

  ];

# app/greet/greet.component.css

.greeting-container {

display: flex;

flex-direction: column;

align-items: center;

gap: 10px;

}

input {

padding: 5px;

font-size: 1rem;

}

button {

padding: 5px 10px;

font-size: 1rem;

cursor: pointer;

}

h3 {

color: #2c3e50;

}

# app/greet/greet.component.html

<div class="greeting-container">

<h2>Введите ваше имя:</h2>

<input [(ngModel)]="userName" placeholder="Ваше имя" />

<button (click)="showGreeting()">Say Hello</button>

<div \*ngIf="greeting">

<h3>{{ greeting }}</h3>

</div>

</div>

# app/greet/greet.component.ts

import { Component, OnInit, OnChanges, DoCheck, AfterContentInit, AfterContentChecked,

AfterViewInit, AfterViewChecked, OnDestroy, SimpleChanges } from '@angular/core';

import { FormsModule } from '@angular/forms';

import { CommonModule } from '@angular/common';

import { RouterModule } from '@angular/router';

@Component({

selector: 'app-greet',

templateUrl: './greet.component.html',

standalone: true,

imports: [FormsModule, CommonModule, RouterModule],

styleUrls: ['./greet.component.css']

})

export class GreetComponent implements OnInit, OnChanges, DoCheck,

AfterContentInit, AfterContentChecked,

AfterViewInit, AfterViewChecked, OnDestroy {

userName: string = '';

greeting: string = '';

constructor() {

console.log('Constructor called');

}

ngOnInit(): void {

console.log('ngOnInit called');

}

ngOnChanges(changes: SimpleChanges): void {

console.log('ngOnChanges called', changes);

}

ngDoCheck(): void {

console.log('ngDoCheck called');

}

ngAfterContentInit(): void {

console.log('ngAfterContentInit called');

}

ngAfterContentChecked(): void {

console.log('ngAfterContentChecked called');

}

ngAfterViewInit(): void {

console.log('ngAfterViewInit called');

}

ngAfterViewChecked(): void {

console.log('ngAfterViewChecked called');

}

ngOnDestroy(): void {

console.log('ngOnDestroy called');

}

showGreeting(): void {

this.greeting = `Привет, ${this.userName}!`;

}

}